

Maqueta: Adaptador de teclado del PC

1. Introducción

El teclado convencional del PC se conecta a la unidad central empleando un conector DIN de 5 terminales a 180°, como el utilizado en algunos amplificadores de audio. Actualmente se suele emplear un conector mini DIN como el que utilizaba el PS/2 y la mayoría de los PCs portátiles. Esta maqueta está diseñada para utilizar un teclado convencional con conector DIN grande.

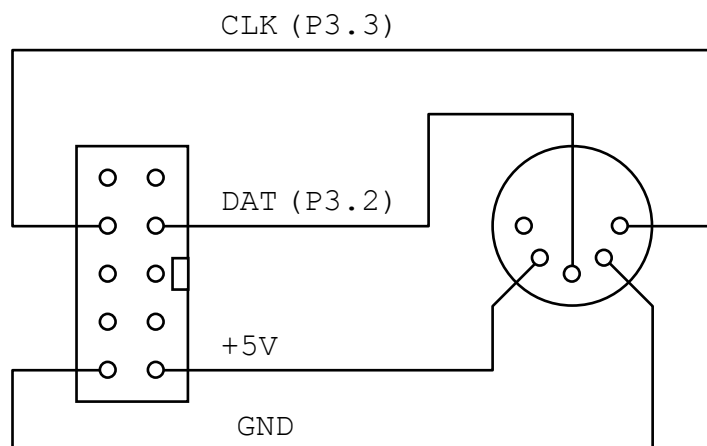
A nivel eléctrico, la conexión se establece mediante 4 hilos que corresponden a masa (GND), alimentación del teclado (+5V), y las señales de reloj (CLK) y datos (DAT).

El enlace es bidireccional y se realiza mediante un protocolo síncrono con señal de reloj separada de los datos. Se trata de un enlace de 2 hilos con un cierto parecido a la conexión I²C.

2. Esquema eléctrico

El teclado opera con niveles TTL, por lo que se puede realizar una conexión directa entre el teclado y la placa de Altair. La tarjeta empleada actúa como un simple adaptador del conector DIN del teclado al conector de 10 terminales utilizado por la placa de Altair.

La figura indica el conexionado.



La señal DAT se conecta a P3.2 (INT 0) y la señal CLK se conecta a P3.3 (INT 1). De esta forma se podrían utilizar las interrupciones externas INT1 e INT2 si se considera conveniente.

3. Desarrollo de la práctica

El objetivo de esta práctica es desarrollar el SW necesario para poder establecer la comunicación entre el teclado y el micro 89C537. La comunicación se debe establecer en ambos sentidos y por iniciativa de cualquiera de las dos partes.

El SW debe ser capaz de detectar la llegada de mensajes procedentes del teclado (pulsación de tecla). La información se debe almacenar en un buffer circular de 16 bytes, y se utilizará un flag *kbhit* que se debe mantener a '1' siempre que el buffer no está vacío. Cuando otro programa quiera saber si se pulsó alguna tecla, consultará este byte (sin modificarlo) y si lo desea retirará byte(s) del buffer empleando un puntero de lectura.

El SW proporcionará una rutina para el envío de órdenes al teclado: reset, encender/apagar leds, etc.

El SW debe escribirse teniendo en cuenta que será integrado en un programa más complejo que realizará otras tareas. Se debe especificar claramente los recursos empleados, incluyendo terminales de puertos, temporizadores, interrupciones y sus prioridades.

El apartado 4 describe los aspectos más importantes del protocolo de comunicación empleado por el teclado del PC.

4. Protocolo de enlace entre el teclado PC/AT y el microprocesador

Este apartado describe de forma resumida el protocolo del teclado. La información se deriva de la que facilita el manual del PC/AT de IBM (modo 11 bits).

En primer lugar se describe la comunicación en forma de comandos. Posteriormente se especifica el formato con el que se envían los comandos y cómo se obtiene el acceso al medio.

4.1. Arranque del teclado

Cuando el teclado recibe alimentación, realiza un proceso de inicialización que se compone de dos tareas denominadas Reset de encendido (POR) y Test Básico (BAT).

El POR puede durar entre 150 ms y 2 s según el teclado.

Al finalizar el POR, se realiza el BAT, que dura entre 300 ms y 500 ms. Durante el BAT se puede observar como los LEDs se encienden y se apagan. El teclado ignora toda actividad en las líneas DAT y CLK hasta que el BAT finaliza.

Si el BAT termina con éxito, el teclado envía al procesador el comando AA (todos los comandos son bytes expresados en hexadecimal) y comienza la exploración de las teclas. Si el BAT detecta fallos, envía el comando FC e inhibe la exploración de las teclas, a la espera de recibir algún comando del procesador.

En cualquier caso, al finalizar el BAT, el teclado consulta las líneas CLK y DAT para poder responder de forma adecuada según el protocolo. El teclado responde a cualquier comando del sistema en menos de 20 ms

4.2. Comandos que puede enviar el procesador al teclado

Se definen 19 comandos, de los cuales se indican solamente los que se utilizan en esta práctica.

Comando	Hex	Observaciones
Set/Reset LEDs	ED	Le sigue un byte indicando el estado en que deben wquedar los LEDS (B0=Bloq.Desp, B1=Bloq.Num, B2=Bloq.Mayús)
Echo	EE	Para pruebas. El teclado responde con EE
Set typematic	F3	Especifica retardo y repetición teclas
Enable	F4	El teclado borra el buffer interno, inicia exploración teclas y contesta con FA
Disable	F5	El teclado inhibe exploración teclas, borra buffer interno, contesta con FA y espera comandos del procesador
Resend	FE	El teclado reenvía el último comando que envió, excepto si él había enviado un Resend
Reset	FF	El teclado responde con FA, comprueba que el procesador pone CLK y DAT a '1' durante al menos 0,5 segundos. Finalmente realiza un BAT

4.3. Comandos que puede enviar el teclado al procesador

Se definen 8 comandos, de los cuales se indican solamente los que se utilizan en esta práctica.

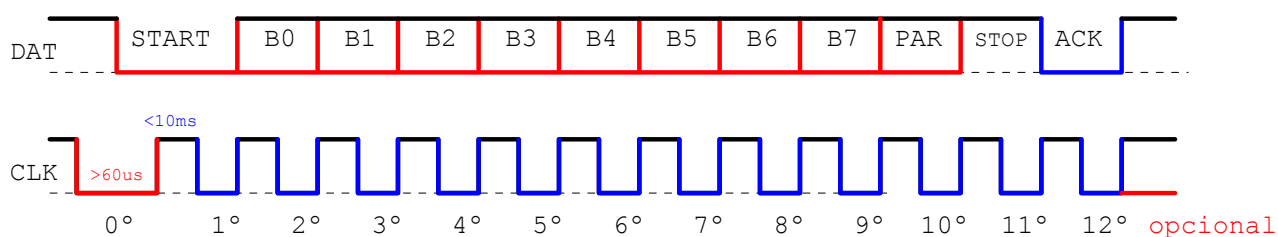
Comando	Hex	Observaciones
Key error/Overrun	00 ó FF	Error en la exploración de las teclas o desbordamiento del buffer de 16 bytes.
El BAT terminó ok	AA	
El BAT dió fallos	FC	
Echoe	EE	Respuesta al comando Echo del sistema
Acknowledge (ACK)	FA	Respuesta a todos los comandos del procesador, excepto a Echo y a Resend.
Resend	FE	El teclado envía este comando si recibe un comando del procesador que no puede reconocer.d

Además, el teclado envía al procesador un código de exploración cada vez que se pulsa o se suelta una tecla. Cuando se pulsa, se envía solamente el código específico de la tecla pulsada (ver apéndice). Cuando se suelta, se envía el mismo código precedido de F0.

4.4. Formato de las tramas

Las líneas CLK y DAT son bidireccionales y se controlan desde ambos extremos (teclado y procesador) con drivers de colector abierto (o drenador abierto). Existen resistencias de pull-up externas (de 2k2 normalmente) que definen un '1' cuando los drivers de ambos extremos están a '1'. En esta práctica, las líneas P3.2 y P3.3 no son realmente salidas a drenador abierto, pero su estructura con pull-up interno débil es compatible. El estado inactivo es '1' en ambas líneas.

La información se envía en tramas de 8 bits de datos por la línea DAT. La figura siguiente muestra una trama que envía información desde el procesador al teclado.



Trama para el envío de datos del procesador al teclado

(los impulsos generados por el procesador en rojo
y los generados por el teclado en azul)

Los impulsos normales de reloj los genera siempre el teclado (azul). Sin embargo, el procesador pone CLK a '0' al principio (impulso 0 en rojo) para inhibir transmisiones del teclado y entonces DAT a '0' para indicar que empieza a transmitir (start). Ahora devuelve CLK a '1' para que el teclado pueda controlar esta línea generando los impulsos normales de reloj (azul). El procesador actualiza el valor enviado en la línea DAT cada vez que el teclado sube la línea CLK a '1'. El primer bit enviado es el B0. Después de B7 va el bit de stop, que es un '1'. Finalmente, el teclado fuerza un '0' en la línea DAT a modo de reconocimiento (ACK). No confundir este bit con el comando ACK descrito en el apartado 4.3.

Si el teclado ve un '0' en lugar de un '1' en la línea DAT cuando corresponde leer el bit de stop (se ha producido un error), continúa emitiendo impulsos de reloj hasta la llegada de un '1' en DAT. De esta forma se produce la *recuperación* de la situación de error.

El teclado captura los datos coincidiendo aproximadamente con el flanco de bajada de CLK. Por lo tanto, el procesador puede actualizar DAT una vez que el flanco de bajada se ha producido. En la figura se dibujan las actualizaciones coincidiendo con el siguiente flanco de subida, pero no es necesario que así sea.

Al finalizar la trama, el sistema puede retener CLK= 0 para impedir que el teclado inicie una transmisión (marcado en la figura como opcional).

Los teclados ensayados generan señales de reloj (CLK) que suelen variar entre 10 y 30 KHz. Según las especificaciones de IBM, deberían respetarse las siguientes cotas temporales:

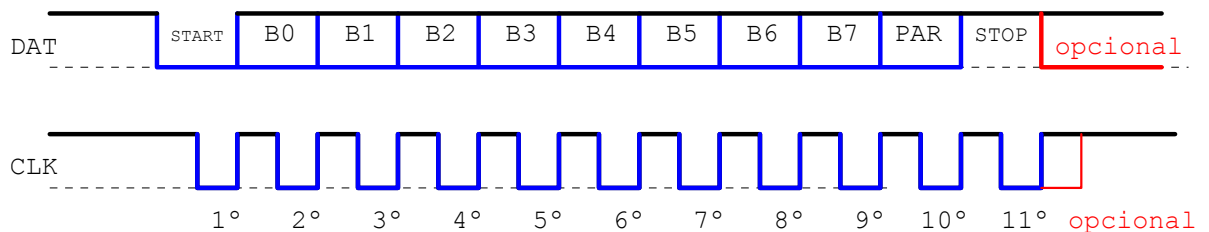
Duración del nivel bajo en la señal de reloj: $T_0 = 30$ a $50 \mu s$.

Duración del nivel alto en la señal de reloj: $T_1 = 30$ a $50 \mu s$.

Intervalo entre el flanco de subida del reloj hasta que el teclado muestrea DAT: 5 a $25 \mu s$.

De los datos anteriores se deduce que el procesador debe actualizar DAT antes de que transcurran $35 \mu s$ desde el flanco de bajada de CLK, es decir $T_0 + 5 \mu s$.

La figura siguiente muestra una trama de envío de información desde el teclado al procesador.



Trama para el envío de datos del teclado al procesador

(los impulsos generados por el procesador en rojo
y los generados por el teclado en azul)

En este caso, todos los impulsos son generados por el teclado, salvo que al finalizar la trama del teclado, el procesador quiera inhibir el envío de nuevas tramas. En este caso, el procesador retiene $CLK = 0$. En realidad puede dejar que suba y esperar hasta un máximo de $50 \mu s$ para bajar CLK nuevamente. El procesador suele realizar esta acción para tener tiempo de procesar el byte recibido antes de que llegue uno nuevo. Es fundamental que el procesador libere CLK en cuanto pueda, para no dejar el teclado bloqueado indefinidamente. También puede ser que el procesador quiera aprovechar para enviar un dato al teclado. Si es así, una vez bloqueado CLK a '0', baja DAT a 0 (start), con lo que ya empieza otra trama (estas opciones aparecen marcadas en la figura como *opcional*).

Aunque la figura muestra que los cambios en DAT coinciden con los flancos de subida de CLK, el teclado no está obligado a hacerlo así. La única obligación del teclado es entregar un valor estable en DAT cuando se produce el flanco de bajada, y lo normal es que el cambio en DAT tenga lugar mientras $CLK = 1$. Obviamente existen unos tiempos de establecimiento (setup) y de retención (hold), que se especifican más adelante.

Los teclados ensayados generan señales de reloj (CLK) que suelen variar entre 10 y 30 KHz. Según las especificaciones de IBM, deberían respetarse las siguientes cotas temporales:

Duración del nivel bajo en la señal de reloj: $T_0 = 30$ a $50 \mu s$.

Duración del nivel alto en la señal de reloj: $T_1 = 30$ a $50 \mu s$.

Intervalo entre el cambio en DAT y el flanco de bajada del reloj: 5 a $25 \mu s$.

Intervalo entre el flanco de subida del reloj y el siguiente cambio en DAT: 5 a 11-5 μ s.
Intervalo entre la subida del CLK nº 11 y la bajada forzada por el procesador: 0 a 50 μ s

De los datos anteriores se deduce que la línea DAT permanece estable desde al menos 5 μ s antes de la bajada de CLK hasta al menos 5 μ s después de la subida de CLK. Si se toma como referencia ($t=0$) el instante del flanco de bajada en CLK, DAT permanecerá estable hasta al menos $t= 30 + 5= 35 \mu$ s (30 μ s es la duración mínima del nivel bajo en CLK).

4.5. Acceso del teclado al medio

Cuando el teclado quiere enviar un dato al procesador, comprueba el estado de las líneas CLK y DAT. Esta verificación no sólo se debe realizar antes de comenzar la trama, sino también durante la trama, al menos cada 100 μ s.

Si CLK= 0, entonces el procesador no puede recibir datos. El teclado debe esperar. El procesador suele poner esta línea a 0 cuando se llena su buffer de recepción, para indicar al teclado que deje de enviar datos.

Si CLK= 1, pero DAT= 0, entonces el procesador va a transmitir un comando al teclado. También en este caso, el teclado debe esperar, pero además debe recibir lo que el procesador le envía.

Si CLK= DAT= 1, entonces el teclado puede enviar datos.

Si durante el envío de la trama el teclado observa que el procesador retiene CLK a '0', debe actuar de la siguiente forma:

- 1) Si la trama se encuentra en sus 10 primeros ciclos, el teclado debe cancelar la transmisión, ya que se ha producido una contienda en el acceso al medio.
- 2) Si la trama ha superado el 10º ciclo, el teclado puede terminar la trama.

4.6. Acceso del procesador al medio

El procesador tiene preferencia en el acceso al medio. Sin embargo debe evitar acceder cuando está recibiendo información del teclado. En el apartado 4.4 se muestra como el procesador toma el control del medio poniendo a '0' la línea CLK durante al menos 60 μ s.

Cuando el teclado recibe un comando, debe responder al procesador en menos de 20 ms, salvo claro está que el procesador se lo impida reteniendo CLK o DAT a '0'.

5. Recomendaciones para realizar la práctica

Aunque el protocolo del enlace no es en realidad complejo, sí es bastante extraño, por lo que suele resultar difícil de entender, al menos en una primera aproximación. Afortunadamente, en esta práctica se aborda solamente la programación del procesador,

no la del teclado. Como se indicó en el apartado 4.6, el procesador tiene preferencia en el acceso al medio, lo que facilita sustancialmente su programación.

5.1 Utilización de interrupciones para la recepción

Según se detalla en el último párrafo del apartado 4.4, la línea DAT es estable durante al menos 35 μ s desde el flanco de bajada en CLK. Por lo tanto, se puede emplear la interrupción externa INT1 (CLK) activada por flanco, para muestrear la línea DAT. Es suficiente con realizar el muestreo en la ISR antes de que transcurran 35 μ s desde que se diparó la interrupción (latencia incluida). La latencia máxima es de 8 μ s incluyendo la llamada a la ISR. La instrucción de salto desde el vector de interrupción precisa otros 2 μ s. El muestreo de la línea DAT debe realizarse en los primeros 25 μ s de la ISR, lo que es perfectamente factible.

Otro aspecto importante es el ritmo máximo al que puede entrar la ISR. En el peor de los casos, el tiempo entre 2 flancos descendentes de CLK será como mínimo $T_0 + T_1 = 60 \mu$ s. Por lo tanto es preciso poder ejecutar la ISR la interrupción en menos de 60 μ s. El código completo de la ISR incluyendo RETI se debe ejecutar en menos de:

$$60 - 10 = 50 \mu$$

Por lo tanto, se debe escribir una ISR optimizada, que realice solamente las tareas básicas de muestrear DAT, verificar que no hay errores y conformar el byte recibido a partir de los bits. La ISR debería activar un flag cuando finalice la recepción de un byte válido, y a ser posible, almacenar el byte en un buffer circular.

5.2 Utilización de interrupciones para la transmisión

El proceso de transmisión parece menos crítico en principio, dado que es el procesador el que lo desencadena. Sin embargo, se debe recordar que incluso en la transmisión, es el teclado el que genera los impulsos de reloj (CLK), y que puede hacerlo a la velocidad máxima permitida, es decir, la misma que se ha considerado para la recepción. En este caso, el procesador dispone de un tiempo máximo de 35 μ s, según el análisis realizado en el párrafo destacado en negrilla en el apartado 4.4. Por lo tanto la transmisión es tan exigente como la recepción y debería realizarse, a ser posible, empleando la misma ISR. Naturalmente, ambos procesos no tendrán lugar simultáneamente.

5.3 Gestión de los bytes recibidos

Desde el programa principal se pueden retirar los bytes almacenados en el buffer circular de recepción, para utilizarlos de forma apropiada. Dado que el teclado no envía datos de forma espontánea, sino solamente cuando el usuario pulsa las teclas (o por efecto typematic), no vale esperar una tasa elevada de bytes recibidos. Sin embargo, hay que contar con la posibilidad de la llegada de varios bytes seguidos, ya que algunos códigos de exploración se componen de varios bytes (ver apéndice).

También es conveniente releer los apartados 4.1 a 4.3 para conocer las respuestas posibles del teclado a los comandos enviados por el procesador (por ejemplo los comando ACK, ECHO, etc.), es decir el protocolo de *alto nivel*.

Apéndice: Códigos de exploración

La siguiente tabla relaciona los códigos de exploración de las teclas con el ASCII correspondiente. Sólo se muestran los ASCII de las teclas alfanuméricas. No se muestran los de las teclas del teclado numérico ni las de control. Cuando una misma tecla representa varios símbolos, sólo se indica el más común. Las letras se indican con el ASCII de la mayúscula. El código de exploración se usa como índice.

Cuando se pulsa una tecla, el teclado envía su código de exploración. Cuando se suelta, envía el mismo código precedido de F0. Es importante recordar que el teclado del PC dispone de un mecanismo de repetición automático denominado typematic, que afecta a casi todas las teclas. De esta forma, si una tecla se mantiene pulsada durante un cierto tiempo, el teclado comienza a enviar códigos de exploración repetidos cada cierto intervalo de tiempo, aunque sólo envía un código de soltar, cuando finalmente se suelta la tecla.

Además, debe tenerse en cuenta que algunas teclas generan códigos de exploración formados por varios bytes.

Esta tabla se facilita a título informativo, para ayudar en la comprobación del SW desarrollado. Una vez que se ha establecido la comunicación con éxito entre el teclado y el microcontrolador, es posible ensayar otras teclas y anotar sus códigos de exploración, aunque no es necesario hacerlo de forma exhaustiva.

```
; Tabla de asciis frente a scancodes
ascii:
db '?','?','?','?','?','?','?','?','?','?','?','?','?','?','?','?' ; 0X
db '?','?','?','?','?','?','Q','1','?','?','?','?','Z','S','A','W','2','?' ; 1X
db '?','C','X','D','E','4','3','?','?','?','?','V','F','T','R','5','?' ; 2X
db '?','N','B','H','G','Y','6','?','?','?','?','M','J','U','7','8','?' ; 3X
db '?','?','?','K','I','O','0','9','?','?','?','?','?','?','?','?','?','?' ; 4X
db '?','?','?','?','?','?','?','?','?','?','?','?','?','?','?','?','?' ; 5X
db '?','?','?','?','?','?','?','?','?','?','?','?','?','?','?','?','?' ; 6X
db '0','?','2','5','6','8','?','?','?','?','?','?','?','?','?','?','?' ; 7X
;
```